IN THE SPECIFICATION

Please amend the Specification as follows:

Page 2, line 4, under the Related Patent Applications section, please amend the paragraph starting there-at as follows:

"This non-provisional United States (U.S.) patent application claims the benefit of and is a continuation application of U.S. Patent Application No. 09/469,961 filed on December 21, 1999 by inventors Edward T. Grochowski, et al., entitled "REPLAY MECHANISM FOR CORRECTING SOFT ERRORS", which claims the benefit of and [This patent application] is a continuation-in-part of U.S. Application Serial No. 08/994,503, entitled "PROCESSOR PIPELINE INCLUDING BACKEND REPLAY", which was filed on December 19, 1997, now U.S. Pat. No. 6,047,370."

Page 4, line 5, please amend the paragraph beginning there-at as follows:

"Another computer system provides execution redundancy using dual execution cores on a single processor chip. This approach eliminates the need for inter-processor signaling, and detected soft errors can usually be corrected. However, the processor employs an on-chip ~~microde~~ microcode to correct soft errors. This approach consumes significant processor area to store the microcode and it is a relatively slow correction mechanism."

Page 8, line 20, please amend the paragraph beginning there-at and continuing to Page 9, line 3 as follows:

"Fig. 2A is a block diagram of one embodiment of processor 110 (processor 210) that supports soft error detection through redundant execution clusters. Processor 210 includes a pair of execution cores 216(a), 216(b) (generically, execution core 216), which are operated in lock step. Each execution core ~~214~~ 216 includes a replay unit ~~270~~ 170 (170(a) and 170(b)) and an execution unit 280 (280(a) and 280(b)). Identical instructions are provided to replay unit ~~270~~ 170 by, e.g. a fetch unit (not shown). Each replay unit ~~270~~ 170 directs the instruction to its associated execution unit 280 and monitors the issued instructions until they retire."

Page 9, line 4, please amend the paragraph beginning there-at as follows:

"Results generated by execution units 280 are compared by check unit 160 and a discrepancy indicates a soft error may have occurred. When a discrepancy is detected, check unit 160 signals an error to replay unit ~~270~~ 170, which reissues selected instructions. If the soft error was transient, e.g. a bit flipped in a logic or control circuit, the discrepancy disappears when the instructions are re-executed. If the discrepancy is not eliminated by re-execution, processor 210 may invoke a back-up recovery mechanism. The discrepancy may persist, for example, if data in a register file or data cache of processor 210 was corrupted by a soft error. For one embodiment of processor 210, check unit 160 invokes a firmware error recovery routine in non-volatile memory ~~140~~ 130 if re-executing instructions a selected number of times fails to eliminate the discrepancy."

Page 9, line 14, please amend the paragraph beginning there-at and continuing to page 10, line 2 as follows:

"Fig. 2B represents another embodiment of processor 110 (processor 220) that supports soft error detection through redundant execution. For the disclosed embodiment of processor 220 only duplicates portions of the processor hardware in back end 118. Protected execution unit 180 includes first and second execution units 280(a) and 280(b). A single replay unit ~~270~~ 170 provides identical instructions to execution units 280 and tracks them until they retire. As for the case of processor 210, processor 220 provides a level of redundant execution that allows soft errors to be detected more easily. However, only the back end stages of processor 220 are duplicated. This reduces the hardware cost for processor 220, but processor 220 may be more susceptible to soft errors in front end 114. As in the embodiment of Fig. 2A, check unit 160 monitors execution units 280 and signals replay unit ~~270~~ 170 when a discrepancy is detected. Processor 220 may also implement a back-up recovery mechanism for those cases in which re-execution does not eliminate the discrepancy."

Page 10, line 21, please amend the paragraph beginning there-at as follows:

"Fig. 3 represents in greater detail one embodiment of processor 210. For the disclosed embodiment, each execution core 216 is represented as a series of stages in an instruction execution pipeline. Each stage corresponds to one or more operations implemented by execution cores 216 to execute their instructions. Alternatively, the pipeline stages may be understood to represent the logic that executes the indicated

operations. Instructions and data are provided to execution cores 216 from a memory system 370. Memory system 370 may represent, for example, main memory 120 and non-volatile memory 130 of Fig. 1. Cache 380 represents a portion of memory system 370 to which results from executed instructions are written. Cache 380 may be located on the same chip as processor ~~100~~ 210 or it may be located on a separate chip."

Page 11, line 14, please amend the paragraph beginning there-at as follows:

"For the disclosed embodiment, results from retired μop(s) are written to cache 380 through retirement channel 364. Because execution cores 216(a), 216(b) operate redundantly, only one of retirement channels 364 needs to update cache ~~280~~ 380. One embodiment of processor 210 may implement a high performance mode in which execution cores 216 operate independently. For this embodiment, both retirement channels 364 are active."

Page 12, line 18, please amend the paragraph beginning there-at and continuing to page 13, line 4 as follows:

"Execution cores 216(a) and 216(b) are synchronized to operate on identical instructions in lock step to support high reliability execution. One embodiment of processor 210 may provide a high performance (HP) mode in addition to the high reliability (HR) mode. In HP mode, execution cores 216(a) and 216(b) operate on different instructions. For example, processor 210 may operate as a single chip symmetric multi-processing (SMP) system in HP mode, with each execution core 216

operating as an independent processor core. Dual mode embodiments of processor are described in U.S. Patent Application Serial No. 09/470,096 ________, entitled "Microprocessor Having a High Reliability Operating Mode" and filed on even date herewith, and U.S. Patent Application Serial No. 09/470.098 ________, entitled "Microprocessor Having a High Reliability Operating Mode" and filed on even date herewith."

Page 14, line 8, please amend the paragraph beginning there-at as follows:

"As discussed above, one embodiment of processor 210 may be switched between a high reliability (HR) mode, in which execution cores 216 operate in lock step, and a high performance (HP) mode, in which execution cores 216 operate on different instruction segments. The ENABLE input to AND gate 430 allows check unit 160 to be disabled when processor ~~300~~ 210 is in HP mode."

Page 15, line 22, please amend the paragraph beginning there-at and continuing to page 16, line 3 as follows:

"For an alternate embodiment of replay unit 170, decoder ~~570~~ 574 may operate on instructions before they are stored in slots 520. For yet another embodiment, fetch unit 570 may provide instruction bundles to replay unit 170, which are then mapped to specific execution units by decoder 574. The extent of DEC stage for the embodiment of processor 210 is indicated in the figure."

Page 16, line 9, please amend the paragraph beginning there-at as follows:

"For the disclosed embodiment of processor ~~110~~ 210, replay unit 170 may be incorporated in the logic associated with DEC stage (Fig. 4) and back-end 580 includes logic associated with REG, EXE, DET, and RET stages. Pointers 530, 540, 550 are updated as instructions are received from FET stage, transferred to REG stage, and retired in RET stage, respectively. For this embodiment, pointer 530 ("head pointer") indicates the latest instruction(s) to enter queue 510, pointer 540 ("tail pointer") indicates the next instruction(s) to be issued to the REG stage, and pointer 550 indicates the next instruction to be retired ("replay pointer") from RET stage. At a given time, the instructions in the slots that follow tail pointer ~~530~~ 540, up to and including the instruction(s) indicated by replay pointer 550, are being executed ("in-flight") in back-end 580. Head pointer 530 is updated when a new instruction enters REG stage, tail pointer 540 is updated when a new instruction enters replay unit 170 from instruction cache 570, and replay pointer 550 is updated when the instruction to which it currently points enters RET stage."

Page 16, line 21, please amend the paragraph beginning there-at and continuing to page 17, line 8 as follows:

"When the disclosed embodiment of processor 110 is operating in redundant mode, check unit 160 signals an error and flushes the back end pipe stages if it detects discrepancy between the execution results in the DET stages of execution cores ~~310(a) and 310(b)~~ 216(a) and 216(b). When control unit 560 detects the error signal, it adjusts tail pointer 530 to indicate the slot currently indicated by replay pointer 550.

This effectively reschedules all un-retired instructions that are currently in the back end of the pipeline for (re)issue to the REG stage. For one execution core/cluster, the instruction(s) indicated by replay pointer 550 is the source of the erroneous execution result, and the instruction(s) in the slots between head pointer 530 and replay pointer 550 follow this error-generating instruction in the back-end of the pipeline. All of these instruction(s) may be flushed from the back end of the pipeline, and reissued by replay unit 170, beginning with the instruction(s) that triggered the error."

Page 19, line 5, please amend the paragraph beginning there-at as follows:

"Replay unit 170 provides a relatively efficient hardware mechanism for correcting soft errors associated with logic, latches, and other storage locations in execution cores ~~310~~ 216. It eliminates the need for providing parity protection for these locations. As noted above, soft errors in certain storage resources can not be corrected by replay unit 170. For example, when a soft error corrupts an operand in one of the data register files, re-executing instructions on the corrupted input data will not alleviate the mismatch between instruction results generated with the corrupted and uncorrupted data. For these and similar errors that can not be corrected through replay, a fall back error correction mechanism may be provided."

Page 19, line 20, please amend the paragraph beginning there-at and continuing to page 20, line 8 as follows:

"For one embodiment of computer system 100, a recovery routine is provided through non-volatile memory 140. Check unit

160 may trigger a machine check that invokes a firmware-based error handling routine. For this embodiment, processor 110 may access an error handling routine when check unit 160 signals an error and replay unit 170 fails to correct it after a specified number of tries. One embodiment of a firmware recovery mechanism operates in conjunction with parity protected storage locations. When replay fails to correct a mismatch in execution results, a recovery routine is implemented to read parity bits associated with the storage structures to locate the error. The storage location that produces the error may be updated with data from the execution core that does not display any parity errors. A firmware based mechanism for processing soft errors is described in U.S. Patent Application Serial No. 09/469,963 _______, entitled "Firmware Mechanism for Correcting Soft Errors" and filed on even date herewith."